PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK DAN INFORMATIKA
UNIVERSITAS MULTIMEDIA NUSANTARA
SEMESTER GANJIL TAHUN AJARAN 2021-2022

IF130 Dasar-Dasar Pemrograman

Pertemuan ke 04 – Pseudocode Struktur Kendali Pengulangan

Yustinus Widya Wiratama, S.Kom., M.Sc., OCA
Anak Agung Ngurah Ananda Kusuma, BE(Hons), MEng, PhD
Putri Sanggabuana Setiawan, S.Kom., M.T.I.

Capaian Pembelajaran Mingguan Mata Kuliah (Sub-CPMK):

1. **Sub-CPMK 05:** Mahasiswa mampu menyusun pseudocode dengan struktur kendali pengulangan (C3)

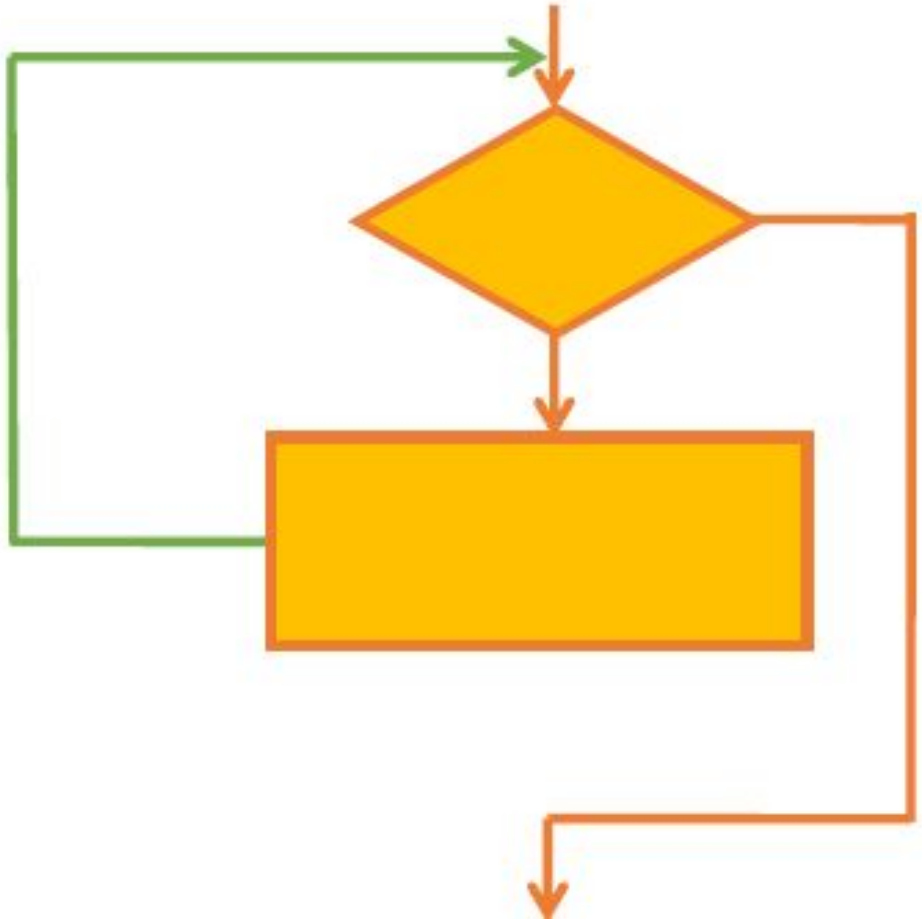
Review

- A loop is a **group of instructions** the computer executes repeatedly while some loop repetition condition remains true.
- 2 kinds of repetition
  1. Sentinel-controlled repetition
  2. Counter-controlled repetition

Outline

1. Pseudocode of repetition control structure
2. Desk checking
3. Exercises

DEFINITION & KIND OF REPETITION

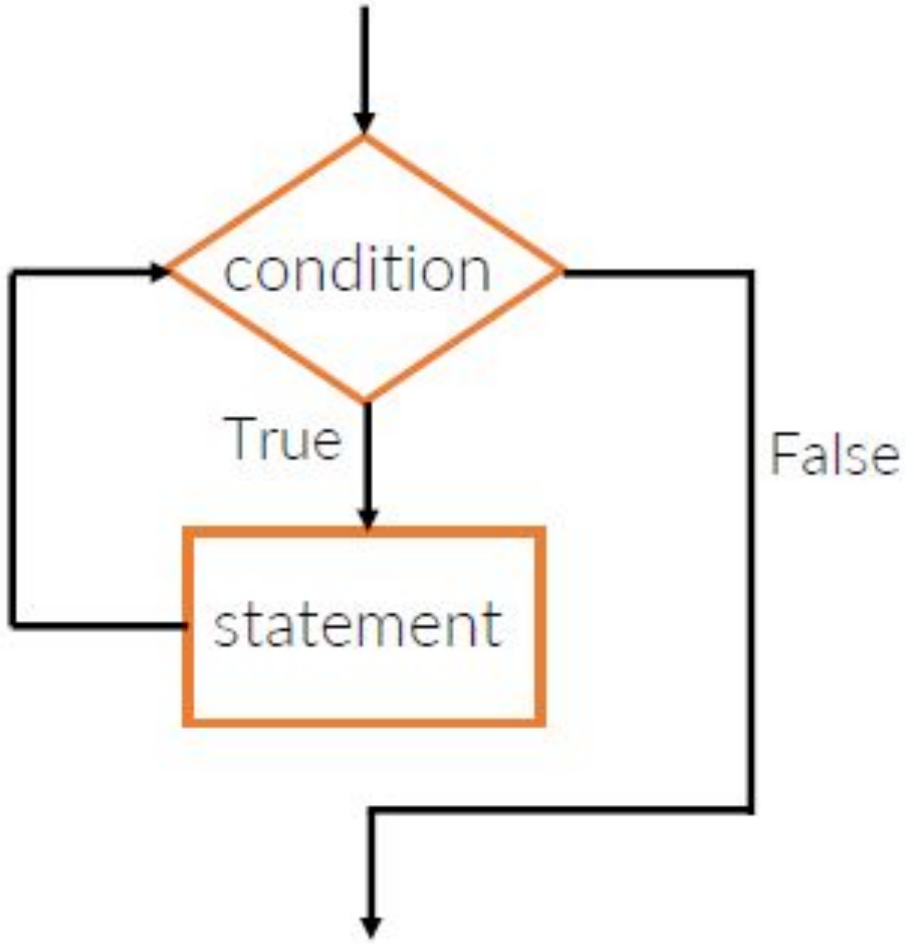


Sentinel-Controlled Repetition

- **While**
- **Do-While**
- **Do-Until**
- Both the **While** and **Do-While** loops cause a statement or set of statements to repeat **as long as** a condition is **true**.
- The **Do-Until** loop causes a statement or set of statements to repeat **until** a condition is **true**.

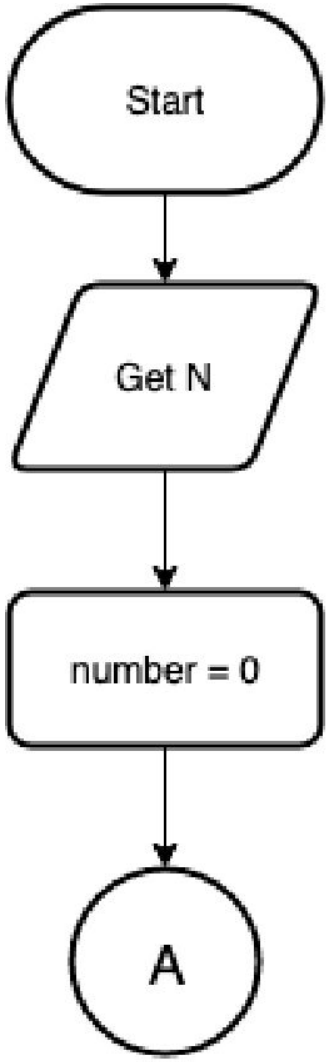
WHILE Loop

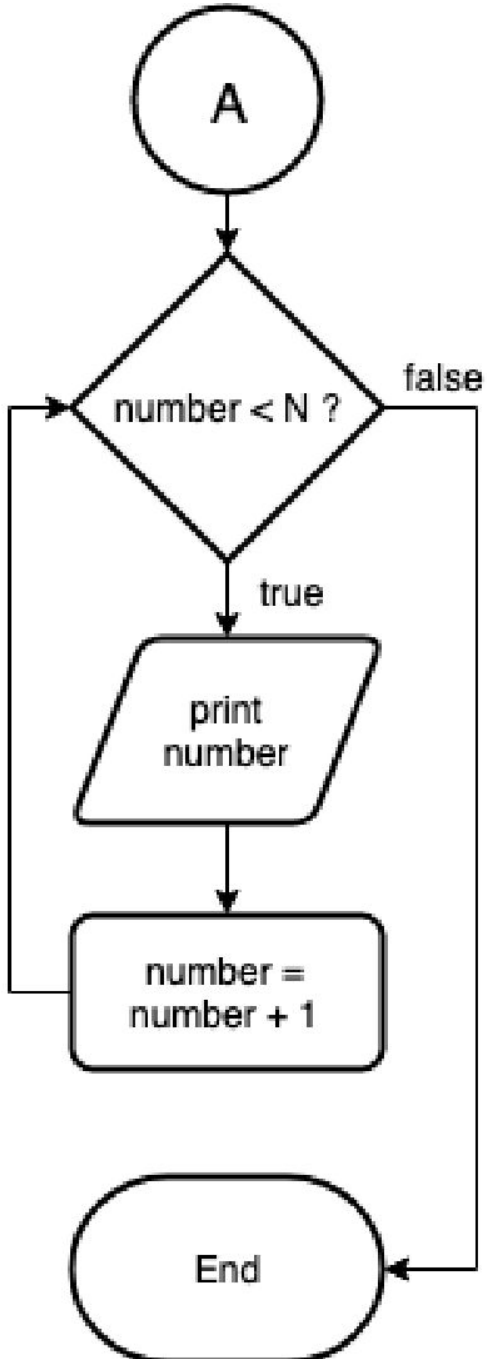
- "While a condition is true, do some task."
- The loop is repeated when the condition is true (when its value is not 0).
- The loop is exited when the condition is false.



WHILE Loop

```
Display_N_numbers
   Prompt for N
   Get N
   Set number to 0
   WHILE number < N
      Print number
      number = number + 1
   ENDWHILE
END
```

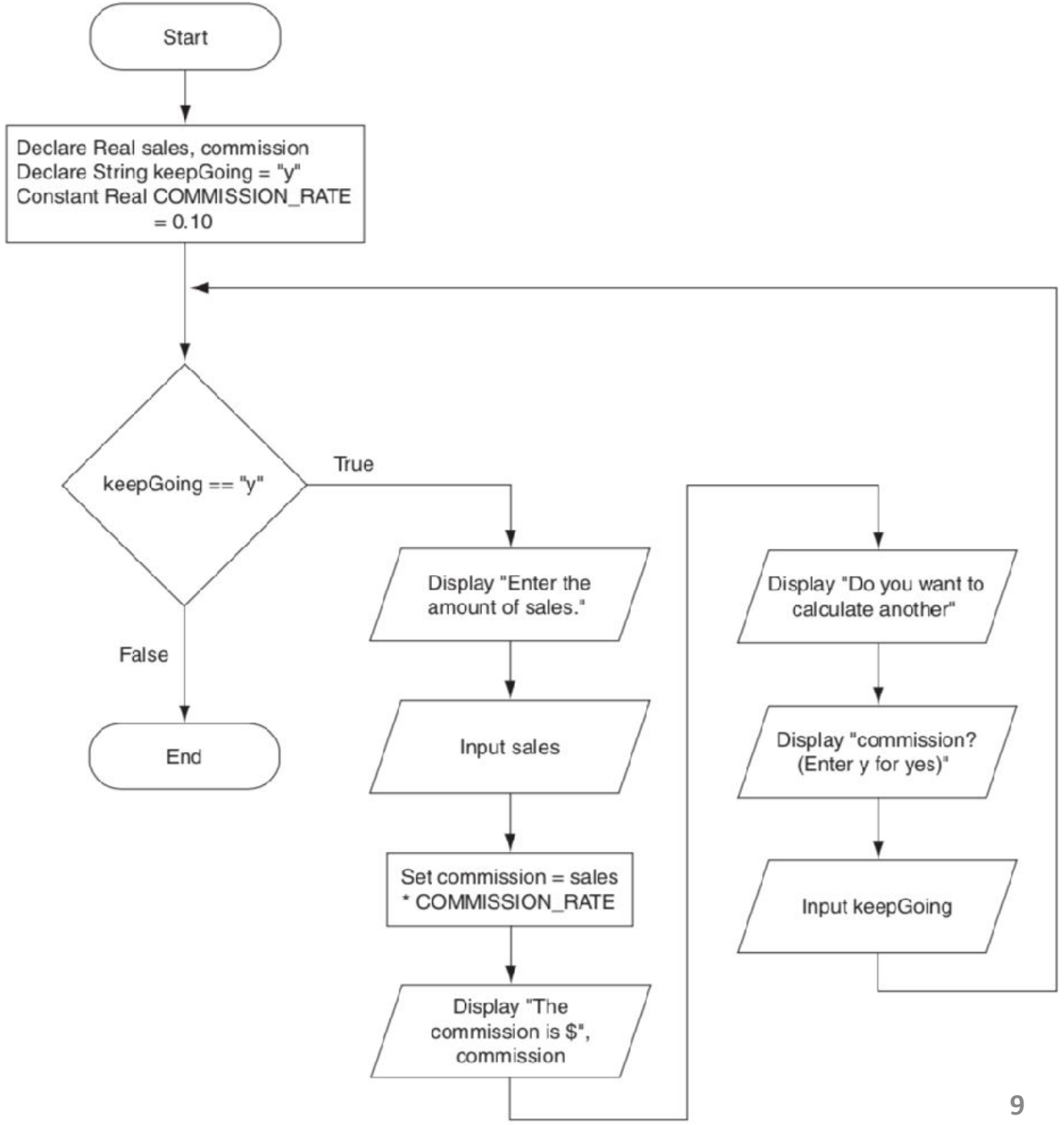




WHILE Loop

Program Output (with Input Shown in Bold)

```
Enter the amount of sales.
10000.00 [Enter]
The commission is $1000
Do you want to calculate another
commission? (Enter y for yes.)
y [Enter]
Enter the amount of sales.
5000.00 [Enter]
The commission is $500
Do you want to calculate another
commission? (Enter y for yes.)
y [Enter]
Enter the amount of sales.
12000.00 [Enter]
The commission is $1200
Do you want to calculate another
commission? (Enter y for yes.)
n [Enter]
```



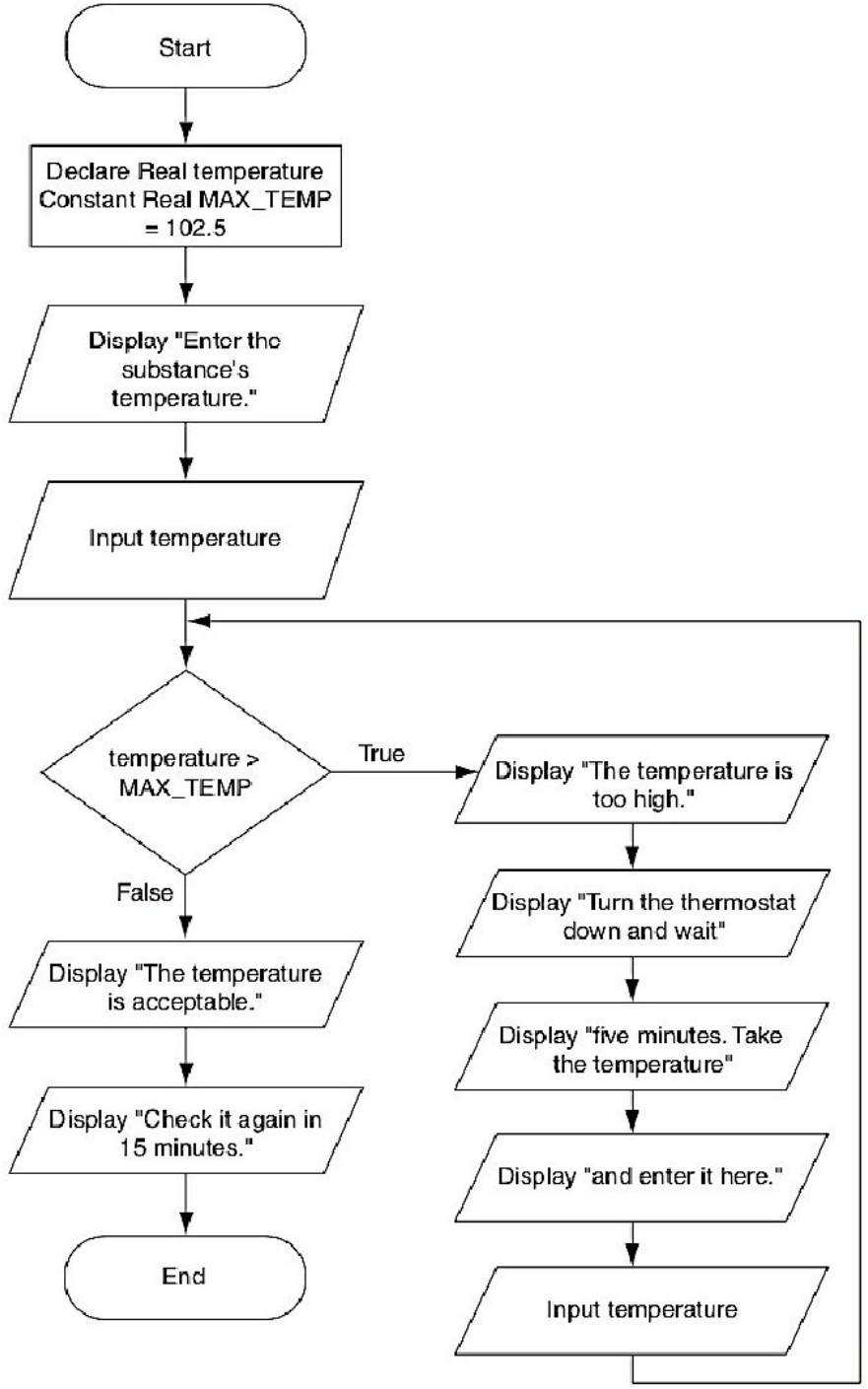
WHILE Loop

Program Output (with Input Shown in Bold)

```
Enter the substance's temperature.
104.7 [Enter]
The temperature is too high.
Turn the thermostat down and wait
five minutes. Take the temperature
again and enter it here.
103.2 [Enter]
The temperature is too high.
Turn the thermostat down and wait
five minutes. Take the temperature
again and enter it here.
102.1 [Enter]
The temperature is acceptable.
Check it again in 15 minutes.
```

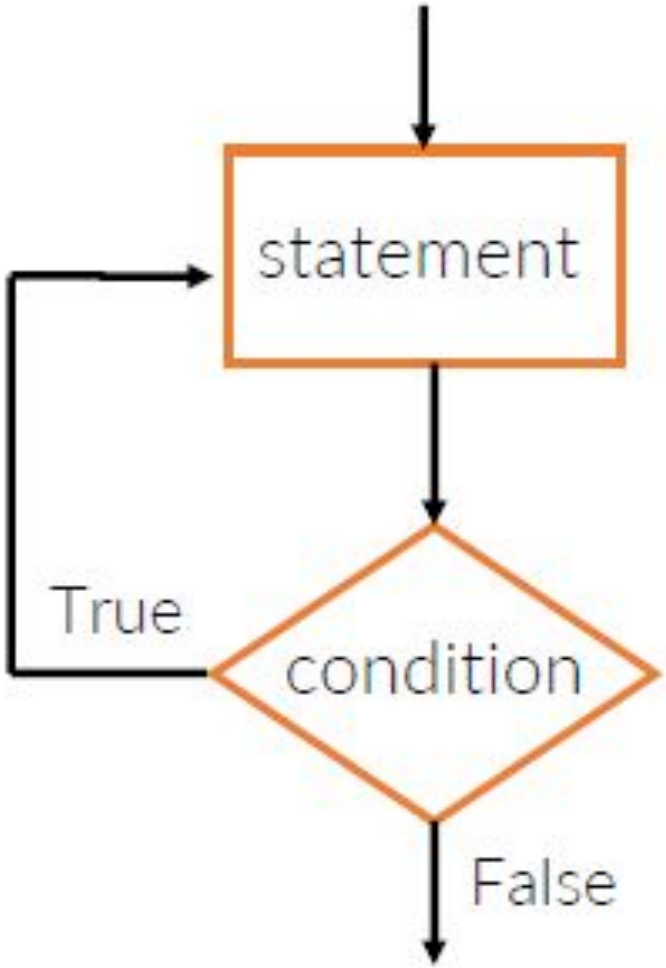
Program Output (with Input Shown in Bold)

```
Enter the substance's temperature.
102.1 [Enter]
The temperature is acceptable.
Check it again in 15 minutes.
```



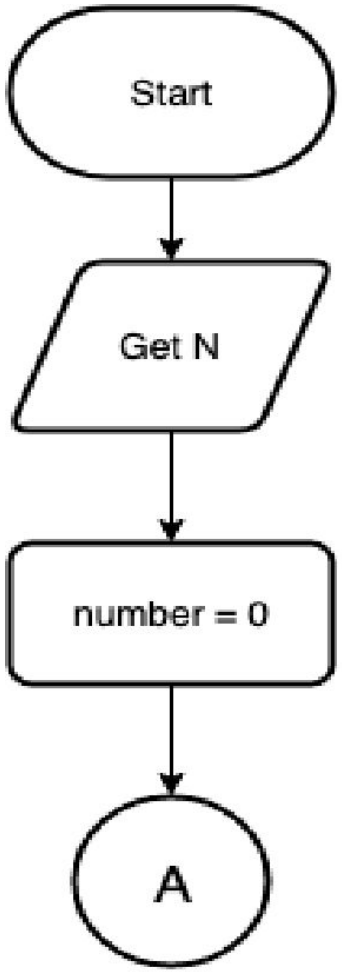
DO-WHILE Loop

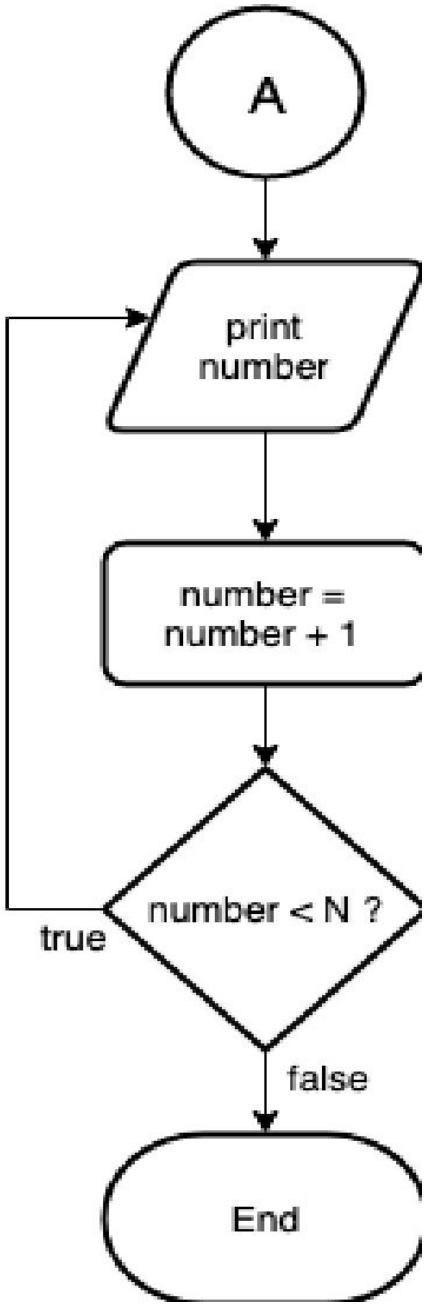
- The **Do-While** loop is a **posttest** loop. This means it performs an iteration before testing its condition.
- As a result, the Do-While loop **always performs at least one iteration**, even if its condition is false to begin with.



DO-WHILE Loop

```
Display_N_numbers
   Prompt for N
   Get N
   Set number to 0
   DO
      Print number
      number = number + 1
   WHILE number < N
END
```





Program output
(with Input Shown in Bold)

10 [enter]

0 1 2 3 4 5 6 7 8 9

DO-WHILE Loop

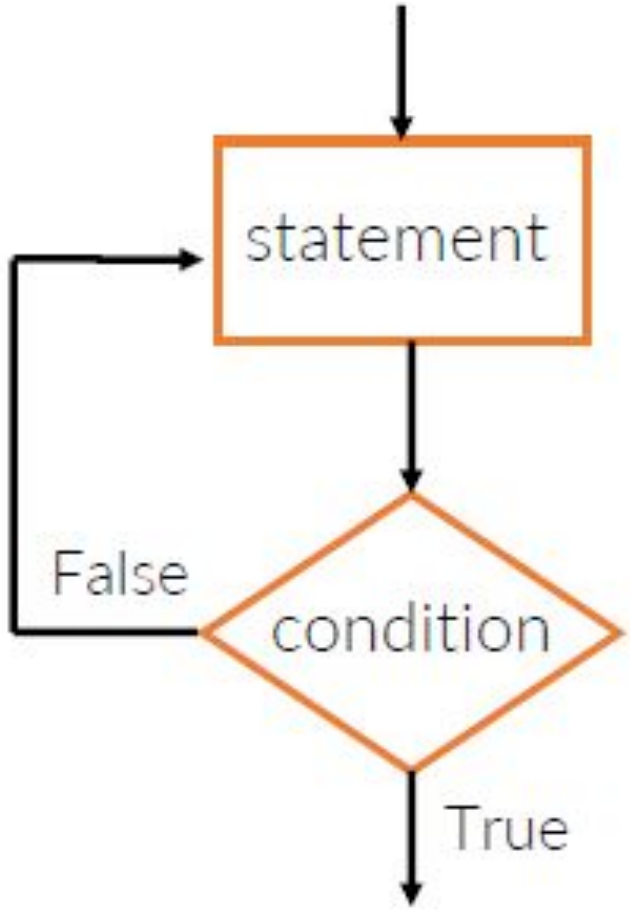
Program Output (with Input Shown in Bold)

```
Enter the amount of sales.
10000.00 [Enter]
The commission is $1000
Do you want to calculate another
commission? (Enter y for yes.)
y [Enter]
Enter the amount of sales.
5000.00 [Enter]
The commission is $500
Do you want to calculate another
commission? (Enter y for yes.)
y [Enter]
Enter the amount of sales.
12000.00 [Enter]
The commission is $1200
Do you want to calculate another
commission? (Enter y for yes.)
n [Enter]
```

```
BEGIN
  DECLARE sales, comm
  DECLARE next
  DO
    PROMPT "Enter the amount of sales."
    GET sales
    comm = sales / 10
    PROMPT "The commission is $", comm
    PROMPT "Do you want ... for yes.)"
    GET next
  WHILE next == 'y'
END
```

DO-UNTIL Loop

- Sometimes, however, it is more convenient to write a loop that iterates **until** a condition is **true**—that is, a loop that iterates as long as a condition is false, and then stops when the condition becomes true .



DO-UNTIL Loop

Program Output (with Input Shown in Bold)

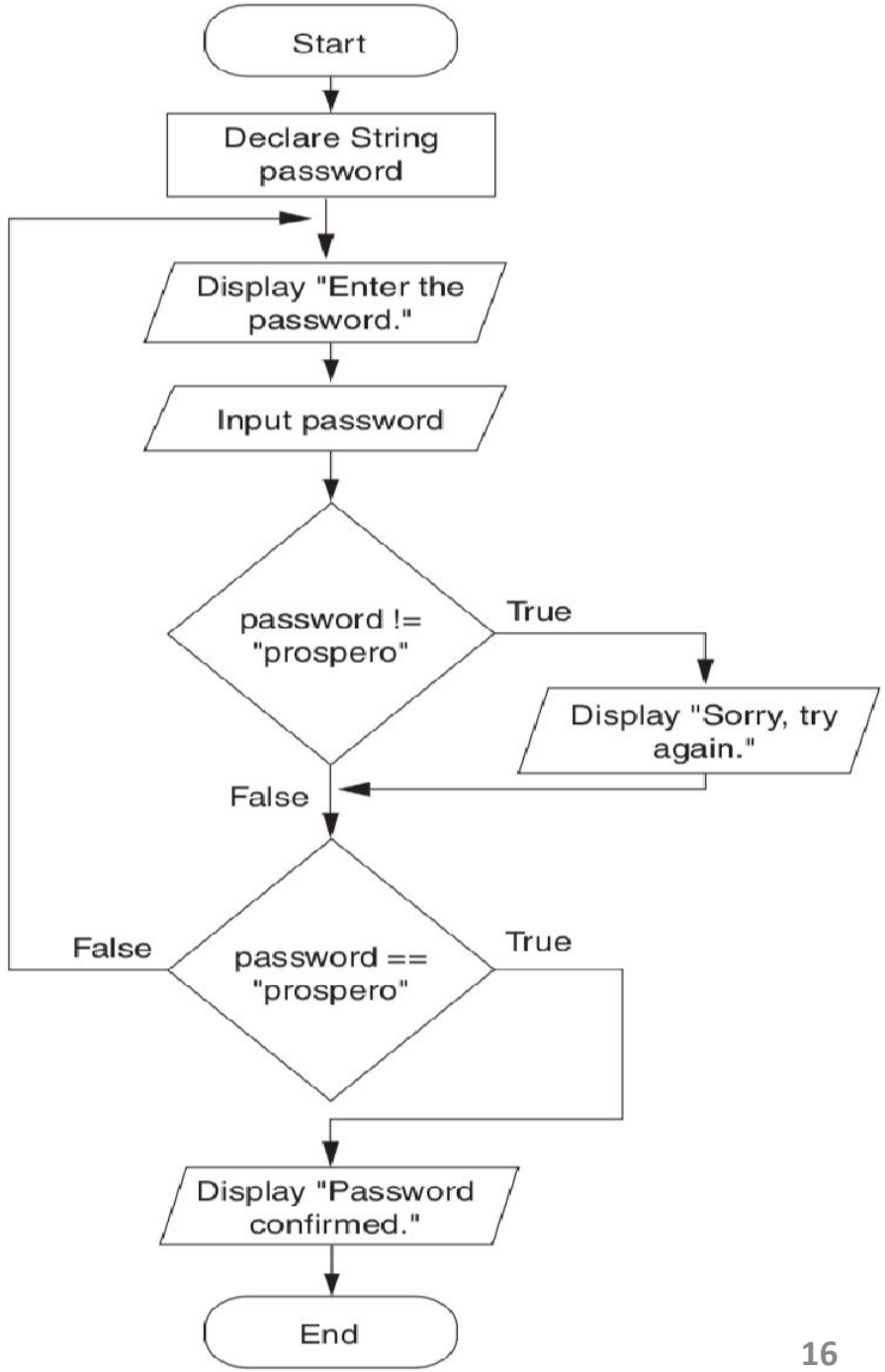
```
Enter the amount of sales.
10000.00 [Enter]
The commission is $1000
Do you want to calculate another
commission? (Enter y for yes.)
y [Enter]
Enter the amount of sales.
5000.00 [Enter]
The commission is $500
Do you want to calculate another
commission? (Enter y for yes.)
y [Enter]
Enter the amount of sales.
12000.00 [Enter]
The commission is $1200
Do you want to calculate another
commission? (Enter y for yes.)
n [Enter]
```

```
BEGIN
  DECLARE sales, comm
  DECLARE exit
  DO
    PROMPT "Enter the amount of sales."
    GET sales
    comm = sales / 10
    PROMPT "The commission is $", comm
    PROMPT "Exit ? (n for no)"
    GET exit
  WHILE exit == 'n'
END
```

DO-UNTIL Loop

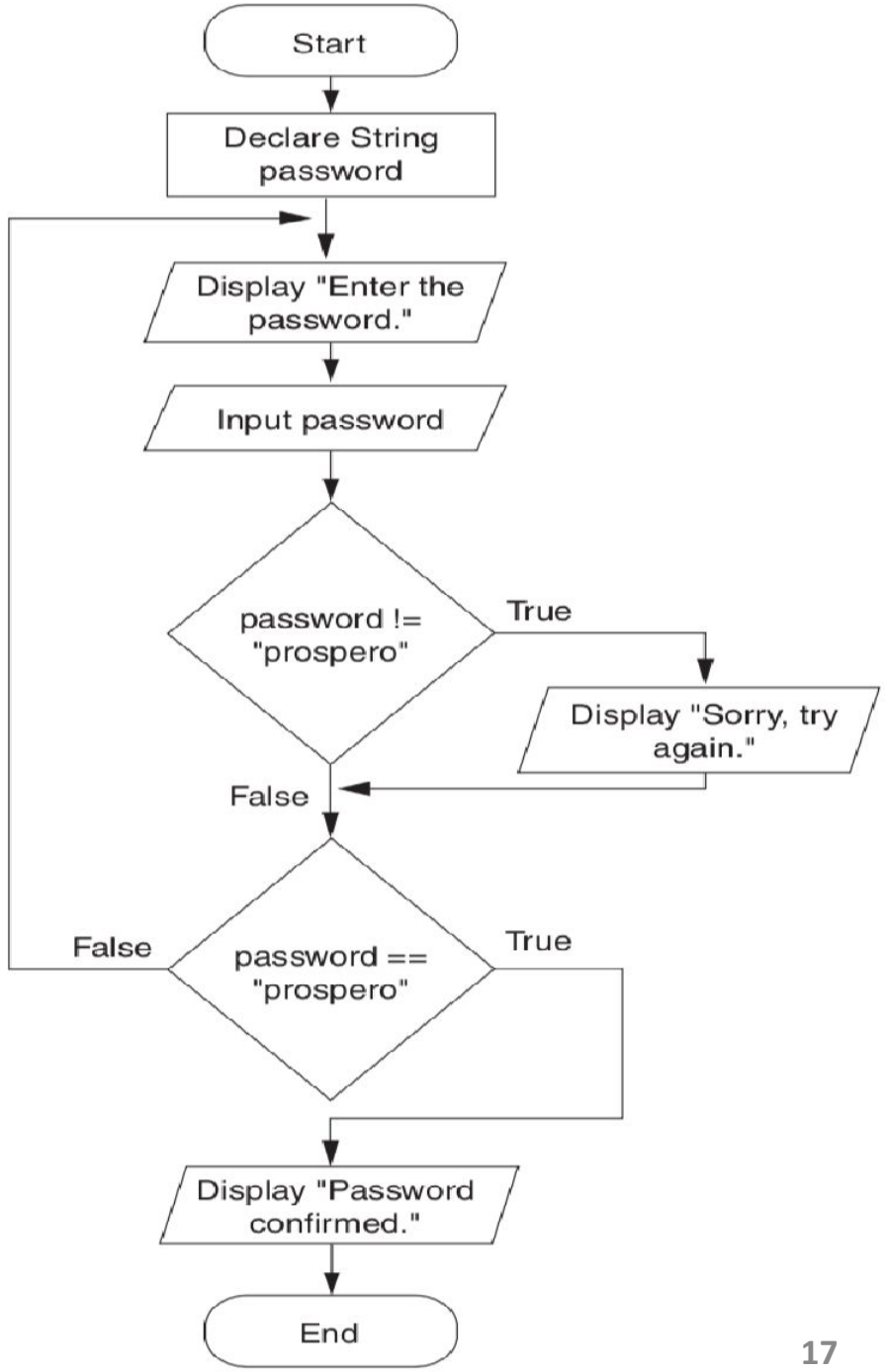
Program Output (with Input Shown in Bold)

```
Enter the password.
ariel [Enter]
Sorry, try again.
Enter the password.
caliban [Enter]
Sorry, try again.
Enter the password.
prospero [Enter]
Password confirmed.
```



DO-UNTIL Loop

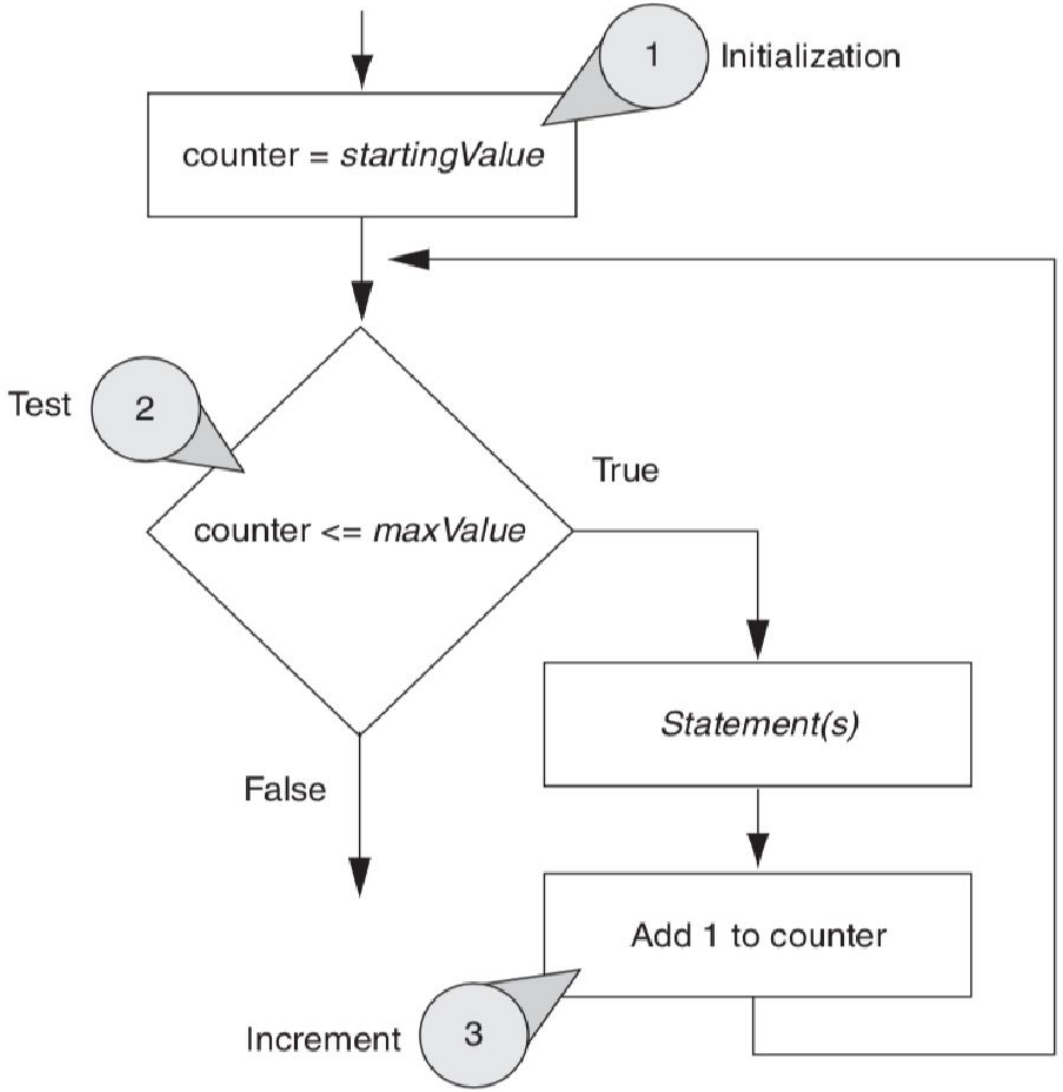
```
Check_password
  Declare string password
  DO
    Display ”Enter the password.”
    Get password
    IF password != “prospero” THEN
      Display “Sorry, try again.
    ENDIF
  UNTIL password == “prospero”
  Display “Password confirmed.”
END
```

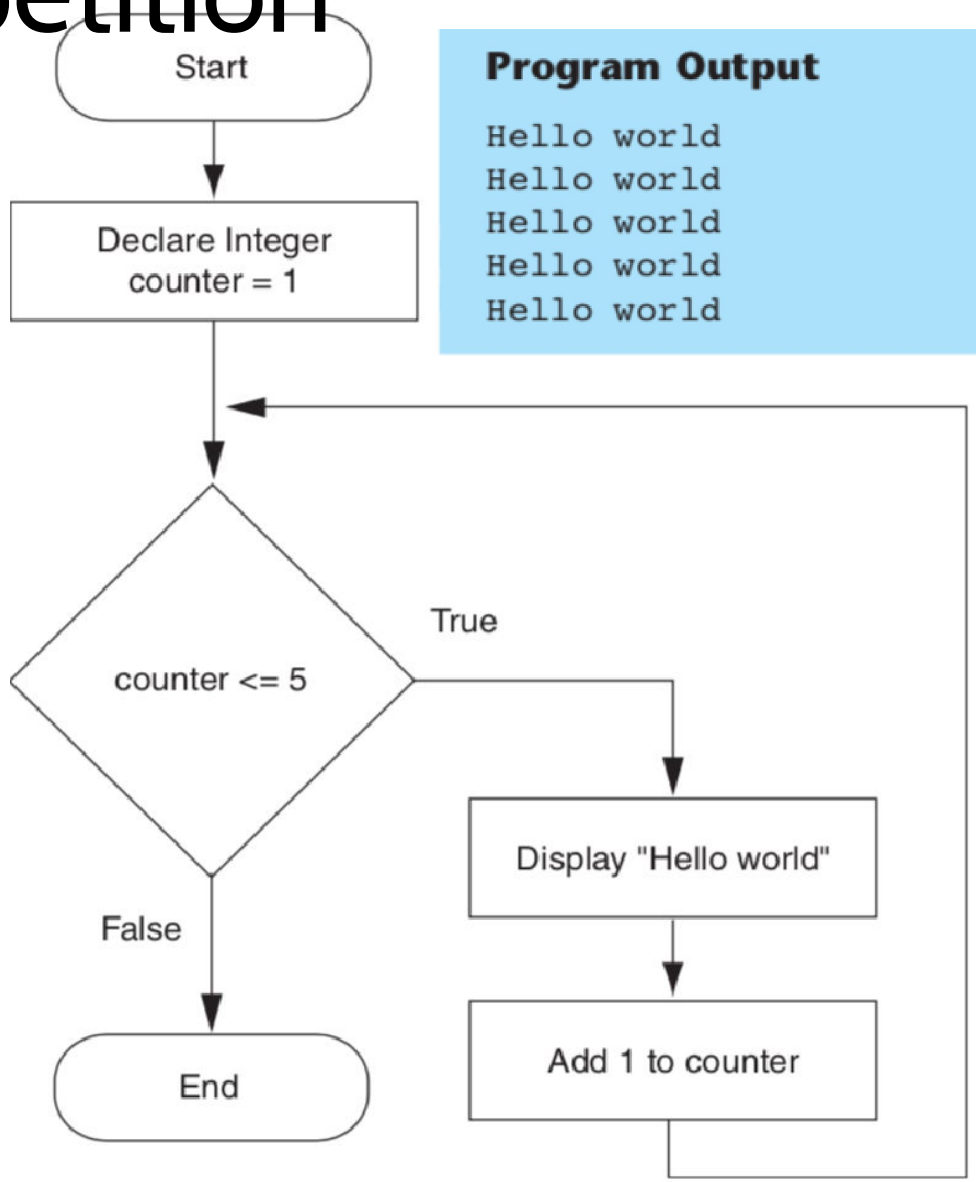


Counter-Controlled Repetition

- Counter-controlled repetition is sometimes called **definite repetition** because we know in advance exactly how many times the loop will be executed.
- This is usually called the **For statement.**
- A loop control variable is used to count the number of repetitions
  1. **Initialization:** Loop control variable is set to an initial value before the while statement is reached
  2. **Testing:** Loop control variable is tested before the start of each loop repetition
  3. **Updating/Increment:** Loop control variable is updated (incremented / decremented) during each iteration

Counter-Controlled Repetition



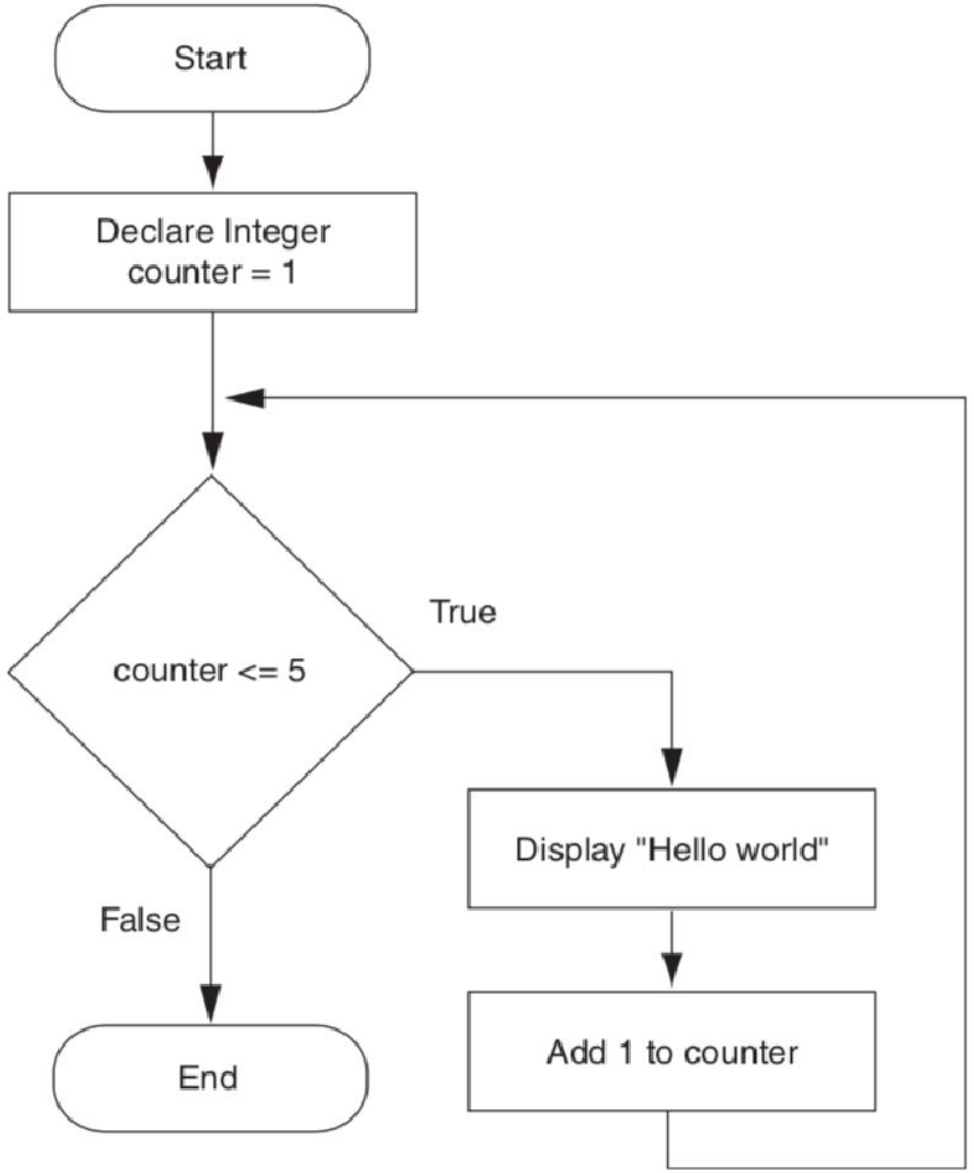


Program Output

```
Hello world
Hello world
Hello world
Hello world
Hello world
```

Counter-Controlled Repetition

```
Counting_integers
  Declare integer counter
  FOR counter = 1 TO 5
    Display “Hello world”
  ENDFOR
END
```



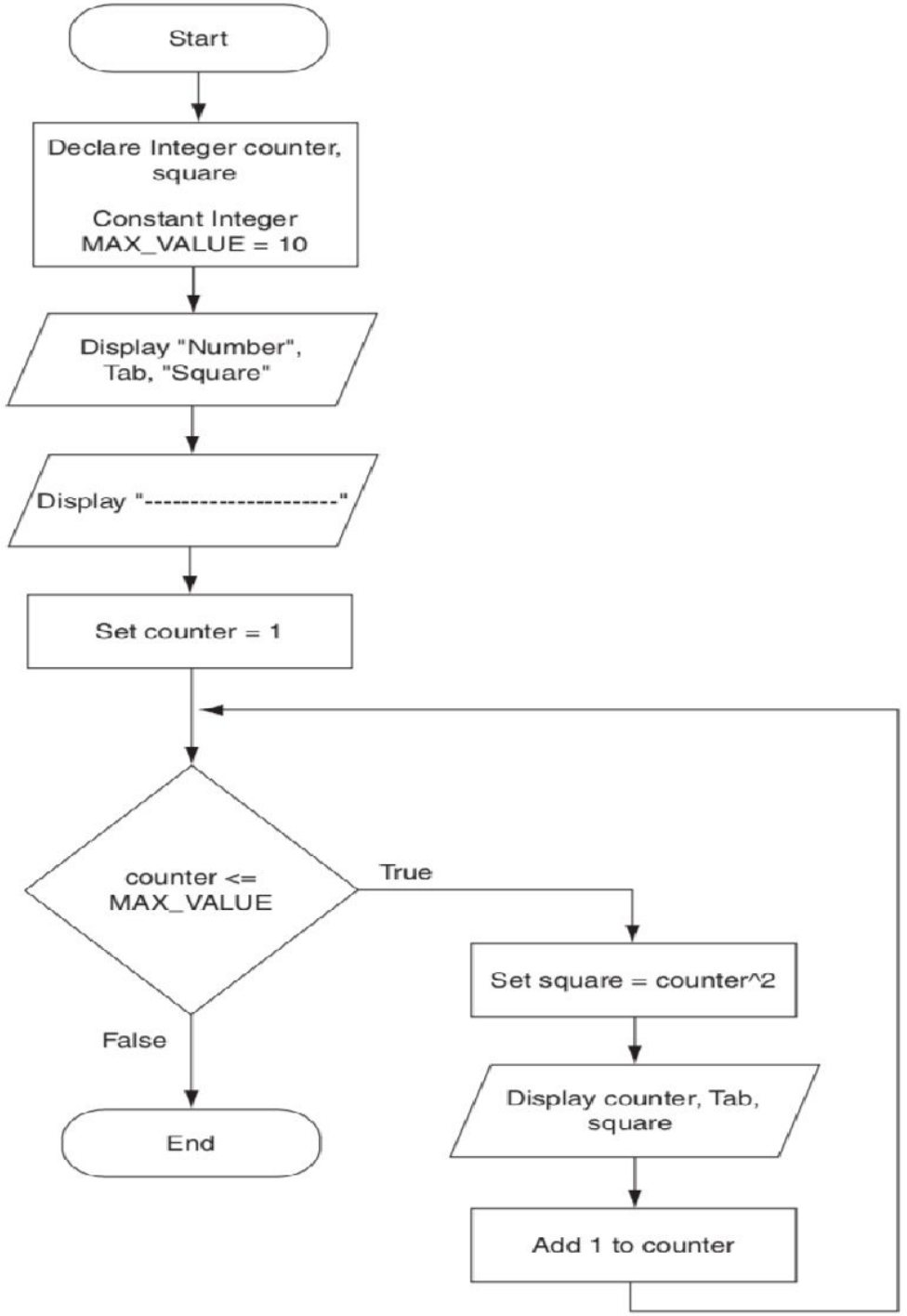
FOR Statement

- In some situations, it is also helpful to use the counter variable in a calculation or other task within the body of the loop.
- For example, suppose you need to write a program that displays the numbers 1 through 10 and their squares.

FOR Statement

Program Output

```
Number          Square
----------------------
   1               1
   2               4
   3               9
   4              16
   5              25
   6              36
   7              49
   8              64
   9              81
  10             100
```



FOR Statement

Program Output

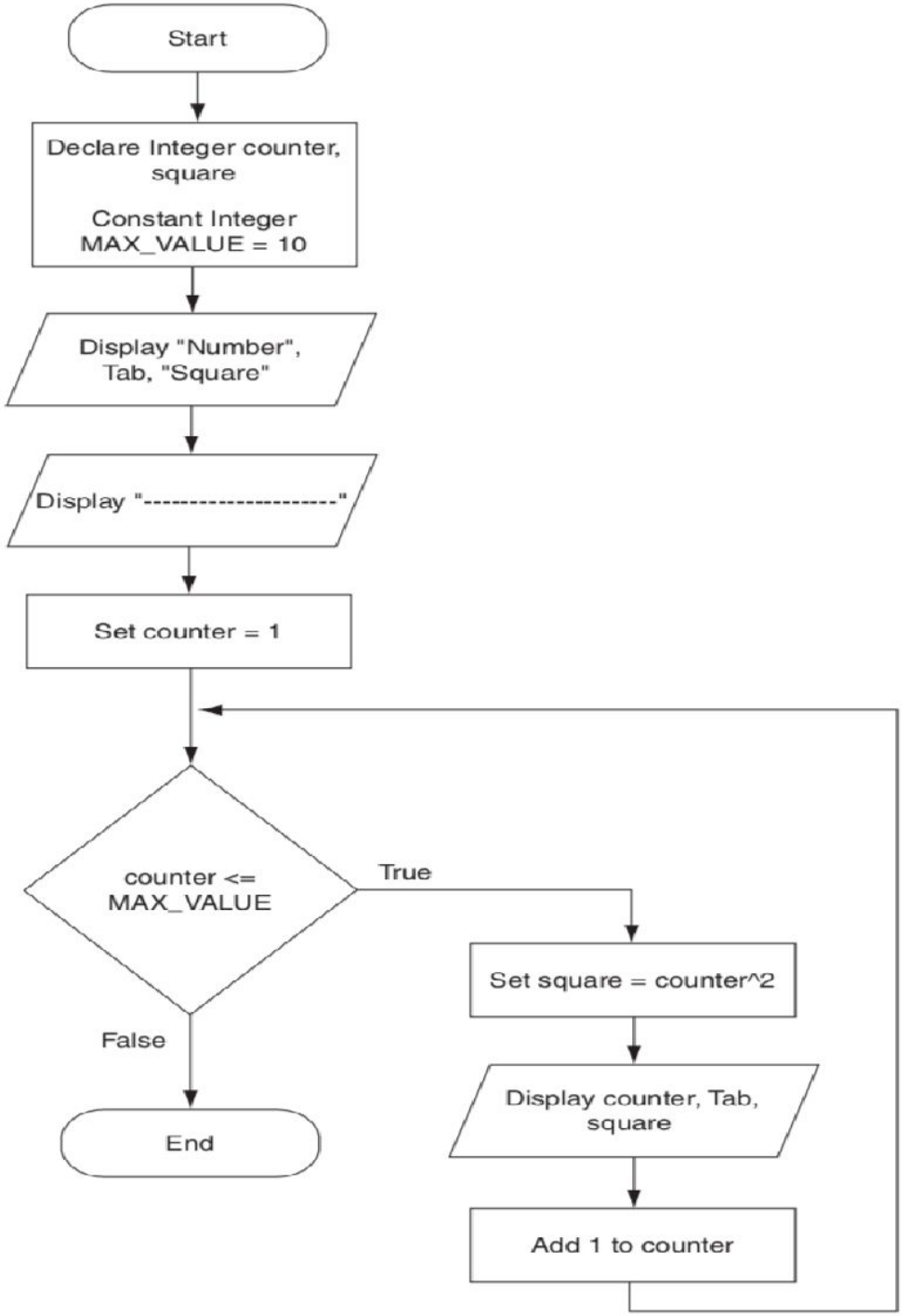
```
Number          Square
----------------------
   1               1
   2               4
   3               9
   4              16
   5              25
   6              36
   7              49
   8              64
   9              81
  10             100
```

```
BEGIN
  DECLARE number, square
  DISPLAY "Number                Square"
  DISPLAY "------------------------"
  FOR number = 1 TO 10
    square = number * number
    DISPLAY number, "                    ", square
    number = number + 1
  ENDFOR
END
```

```
BEGIN
  DECLARE number = 1, square
  DISPLAY "Number                Square"
  DISPLAY "------------------------"
  WHILE number <= 10
    square = number * number
    DISPLAY number, "                    ", square
    number = number + 1
  ENDWHILE
END
```

FOR Statement

```
Square_integers
  Declare integer counter , square
  Set MAX_VALUE to 10
  Display “Number”, Tab, “Square”
  Display “--------------”
  FOR counter = 1 TO MAX_VALUE
    Set square to counter*counter
    Display counter, Tab, square
  ENDFOR
END
```



Practice 1

- Design an algorithm in **pseudocode** which displays the numbers 1 through the maximum value (user input) and their squares.
- Maximum value = 5

Number	Square
1	1
2	4
3	9
4	16
5	25

Practice 2

- Design an algorithm **in pseudocode** that print the following sequence of values

```
20 14 8 2 -4 -10
```

Practice 3

- Design an algorithm **in pseudocode** that print the following sequence of values

```
19 27 34 40 45
```

Practice 4

- The factorial function is used frequently in probability problems. The factorial of a positive integer n (written n! and pronounced “n factorial”) is equal to the product of the positive integers from 1 to n. Write in a **pseudocode** that evaluates the factorials of the integers from p to q (p and q are inputted by user). The screen dialogue should appear as follows:

```
1 5
1! = 1
2! = 2
3! = 6
4! = 24
5! = 120
```

```
3 8
3! = 6
4! = 24
5! = 120
6! = 720
7! = 5040
8! = 40320
```

NEXT WEEK'S OUTLINE

1. Definition of modular programming
2. Modular flowchart
3. Modular Desk checking
4. Exercises

REFERENCES

- Gaddis, Tony, 2019, Starting out with programming logic & design, Fifth edition, Pearson Education, Inc.

Visi

Menjadi Program Studi Strata Satu Informatika **unggulan** yang menghasilkan lulusan **berwawasan internasional** yang **kompeten** di bidang Ilmu Komputer (*Computer Science*), **berjiwa wirausaha** dan **berbudi pekerti luhur**.

INFORMATIKA
UMN
UNIVERSITAS MULTIMEDIA NUSANTARA

Misi

1. Menyelenggarakan pembelajaran dengan teknologi dan kurikulum terbaik serta didukung tenaga pengajar profesional.
2. Melaksanakan kegiatan penelitian di bidang Informatika untuk memajukan ilmu dan teknologi Informatika.
3. Melaksanakan kegiatan pengabdian kepada masyarakat berbasis ilmu dan teknologi Informatika dalam rangka mengamalkan ilmu dan teknologi Informatika.